signal at each of the intervals during the course of a frame of audio. The digital data **408** may additionally group one or more frames of data into packets prior to encryption.

The client **400** further includes an encryption unit **410**, configured to encrypt the digital data **408** according to the encryption mechanism or scheme of a distributed communication system. For example, the encryption unit **410** may utilize an additive homomorphic encryption scheme using the secret key of the client **400**. The encryption unit **410** may be any suitable encryption unit, computer system, processor, or microprocessor capable of performing real-time data encryption of the digital data **408** according to the requirements of the distributed communication system. The encryption unit **410** generates an encrypted data stream **412** corresponding to the digital data **408** using a suitable encryption scheme or algorithm. The encrypted data stream **412** may be represented as a vector of values corresponding to each value of the digital data **408**. The particular encryption scheme or algorithm may be agreed upon by each of the clients engaging in a communication session prior to initiating the communication session.

The client **400** is in electronic communication with the central communication hub or mixer **414** by way of communication channel **416**, and transmits the encrypted data stream **412** to the mixer **414**. The mixer **414** similarly receives encrypted data streams from a plurality of other clients, and combines the encrypted data streams to generate an encrypted result data stream **418**.

The mixer **414** then provides the encrypted result data stream **418** back to the client **400** by way of communication channel **420**. The encrypted result data stream **418** may be represented as a vector of values corresponding to the values of the vectors of the encrypted data streams of the clients. The encrypted result data stream **418** may include a combination of all of the encrypted data streams provided by all of the clients during the communication session, or may include only some of the encrypted data streams. For example, in some embodiments, a particular client may not receive their own encrypted data stream as part of the encrypted result data stream.

The client **400** further includes a decryption unit **424**, which decrypts the encrypted result data stream **418** to generate a decrypted data stream **426**, according to the encryption scheme agreed upon by the clients or according to the design and function of the distributed communication system. The client **400** further includes a decoder **428**, which receives the decrypted data stream **426**, and decodes the decrypted data stream **426** to an appropriate format. For example, the decoder **428** may decode the decrypted data stream **426** to an analog audio signal that can be played back using a player or playback unit **430** to the user of the client terminal **400** using a speaker **432** of the client terminal **400**. In the case of non-voice data communication, the decoded data is send to a desired destination, without any play back.

In some embodiments, a NTRU algorithm is used as a representational additive homomorphic encryption scheme which provides encryption and decryption functions.

FIG. **5** illustrates an example of a block diagram of a communication hub or mixer according to embodiments of the present invention. As shown in FIG. **5**, the communication hub or mixer **500** receives the encrypted data streams **110***a*-**110***d* from each of the client terminals **102***a*-**102***d*, respectively, through mixer inputs **502***a*-**502***d*. Next, one or more key switch circuits perform a key switch operation on the encrypted data streams **110***a*-**110***d* using a corresponding mixer-to-client key switch hint. For example, a key switch circuit **504***a* of the mixer **500** performs a key switch operation on the encrypted data stream **110***a* received from the client terminal **102***a* using the corresponding client-to-mixer key switch hint received from a trusted third party to generate an encrypted data representation **506***a*.

The key switch operation may then be defined as the function: KeySwitch(c1, a12): c2=a12*c1 mod q. The input ciphertext c1 may be encrypted via some key f1 and the ciphertext may be represented, for example, as a vector of integers. The input key switch hint a12 may also be represented as a vector of integers and may be a hint which is used to switch ciphertexts encrypted under f1 to ciphertexts encrypted under f2. The key switch operation may include performing an element-wise multiplication of the input ciphertext and the input key switch hint, modulo an integer q which is pre-shared with the mixer. Thus, the output may be a ciphertext c2 which is an encryption of the data encrypted in c1, but accessible by a holder of the key f2.

Similarly, key switch circuits **504***b*-**504***d* perform a key switch operation on each of the other respective encrypted data streams using the corresponding client-to-mixer key switch hints received from the trusted third party, to generate encrypted data representations **506***b*-**506***d* corresponding to the client terminals **102***b*-**102***d*, respectively. The key switch circuits **504***a*-**504***d* may be separate components or the same component, and in one embodiment, the key switch circuits **504***a*-**504***d* are part of a computer system or processor of the mixer **500**.

The mixer **500** then combines or adds the encrypted data representations **506***a*-**506***d* using a suitable summation scheme. For example, in one embodiment illustrated in FIG. **5**, a summation of the encrypted data representations **506***a*-**506***d* may be performed in a tree fashion, in which the encrypted data representation **506***a* is combined with the encrypted data representation **506***b* using an adder **508***a* to generate a combined data set **510***a*, and the encrypted data representation **506***c* is combined with the encrypted data representation **506***d* using an adder **508***b* to generate a combined data set **510***b*. Next, the combined data set **510***a* is mixed with the combined data set **510***b* using an adder **512** to generate a composite data set **514**.

The addition process may be represented, for example, by the equation Addition(c',c): c"=c'+c mod q where two input ciphertexts of the same dimension c' and c are represented as equal-length integer vectors. The output is the ciphertext c" which is the element-wise addition of c' and c, mod an integer q which is pre-supplied to the mixer. This process may be repeated for every addition step in the mixer.

The composite data set **514** is then provided to a key switch circuit **516** of the mixer **500** to perform a second switching operation on the composite data set **514** to generate an encrypted result data stream **112***a* using the mixer-to-client key switch hint **210***a*. The encrypted result data stream **112***a* is then transmitted to the client terminal **102***a*, where the encrypted result data stream **112***a* can be decrypted using the client private key **202***a*.

The above process is also performed for each of the other client terminals **102***b*-**102***d*, such that each of the client terminals **102***a*-**102***d* receives an encrypted result data stream **112***a*-**112***d*, respectively, which are a composite of the encrypted data streams **110***a*-**110***d*. In the embodiments illustrated in FIG. **5**, encrypted result data stream **112***a* provided to the recipient client terminal **102***a* includes the encrypted data stream **110***a*. Thus, the mixer **500** receives encrypted input data from each of the client terminals and provides a common output.

In some embodiments, the encrypted result data stream **112***a* may include a composite of only the encrypted data